

JC542 U.S. PTO  
 09/672172  
 09/27/00

*Transmitted with the patent application are the following:*

### Fee Calculation:

	Claims		Extra	Rate	Fees
<i>Basic Fee</i>					<b>\$690.00</b>
<i>Total Claims</i>	<b>24</b>	<b>-20 =</b>	<b>4</b>	<b>x \$18.00</b>	<b>\$ 72.00</b>
<i>Independent Claims</i>	<b>3</b>	<b>-3 =</b>	<b>0</b>	<b>x \$78.00</b>	
<i>Multiple Dependent Claims</i>				<b>+\$260.00</b>	
				<b>TOTAL</b>	<b>\$762.00</b>

☒ Please charge my Deposit Account No. 09-0460 in the amount of \$ 762.00. **A duplicate copy of this sheet is attached.**

☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit **Account 09-0460. A duplicate copy of this sheet is attached.**

☒ Any filing fees under 37 CFR 1.16 for the presentation of extra claims.

☒ Any patent application processing fees under 37 CFR 1.17.

Respectfully submitted,  
Curt Cotner et al.

I hereby certify that the above paper/fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date of Deposit: September 27, 2000

Person Mailing paper/fee: Jeanette Berry Durbin

Signature: Jeanette Baker

Prentiss W. Johnson,  
Registration No. #33,123  
Attorney for Applicant(s)  
Telephone (408) 463-5673  
IBM Corporation  
Intellectual Property Law  
555 Bailey Avenue (J46/G467)  
San Jose, CA 95141-1003

*Parthiv Nayan Sharma*

Express mail Label #: EL290558674US

## **SPECIFICATION**

IBM Docket No. STL9-2000-0080US1

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that We, Curt L. Cotner of Gilroy, California and citizen of the United States, Laurence E. England of Morgan Hill, California and citizen of the United States, Howard J. Glaser of San Jose, California and citizen of the United States, and Howard M. Hess of Evanston, Illinois and citizen of the United States have invented new and useful improvements in

### **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**

of which the following is a specification:

1  
2  
3       **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR**  
4       **PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM**  
5       **CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**  
6  
7  
8  
9  
10

11           A portion of the Disclosure of this patent document contains material which is subject  
12 to copyright protection. The copyright owner has no objection to the facsimile reproduction by  
13 anyone of the patent document or the patent disclosure, as it appears in the Patent and  
14 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates in general to software re-engineering, and more particularly to identifying computer program code which is a candidate for Web enablement or stored procedures.

### 2. Description of the Related Art

A large enterprise may have a significant volume of computer application code, the enterprise's code inventory. Yet the enterprise's programming skill base to maintain, manage, enhance, or re-implement that code inventory may be dwindling. Relative to re-implementation, not all application programs are candidates for conversion to a Web interface or a database stored procedure. Many, if not most, applications do not possess the necessary characteristics for such a conversion. Discovering those existing opportunities to easily access a function from the Web or convert a function to a database stored procedure may be difficult due to the labor intensive nature of the conventional manual techniques. Conventional methods have failed to provide adequate solutions to these problems. Thus, there is a clearly felt need for a method of, system for, article of manufacture for, and computer program product for identifying computer program code which is a candidate for Web enablement or stored procedures.

## SUMMARY OF THE INVENTION

The present invention comprises a method, system, computer program product, and article of manufacture for identifying computer program code which is a candidate for Web enablement or stored procedures. Source code corresponding to computer program code is scanned and parsed to determine static information concerning the computer program code. The static information is stored in a database. Dynamic information concerning the computer program code during an execution of the computer program code is also collected and stored in the database. Responsive to the static information and dynamic information stored in the database, relationships and dependencies are then developed and stored in the database. The database may then be queried to produce a set of potential candidates of computer program code meeting a constraint of the query. If insufficient candidates are returned by the query, then the query constraint may be relaxed, and the query repeated.

A candidate for re-implementation as a database stored procedure call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a number, above a specified first threshold, of calls to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a number of calls, above a specified second threshold, by another portion of computer program code.

A candidate for re-implementation as a Web-enabling interface call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program contains a transaction and does not contain screen output related program code.

The placement of this information capture, classification, and relationship/dependency discovery into a database, the Application Knowledge Base, that can be queried may provide significant advantages to a user. For an enterprise, this process may be performed across the

1 entire enterprise, encompassing all of the enterprise's inventory of application programs. The  
2 details of the enterprise's application program inventory may be saved and queried in the  
3 application knowledge base. These details may include system and sub-system specific  
4 information. In addition to queries for identification of candidate computer code, the  
5 application knowledge base may simply be queried for application characteristics.  
6

7 Unlike simple conventional string matching or pattern matching approaches, the present  
8 invention discovers relationships and dependencies within the computer program code. For  
9 example, a COBOL dynamic CALL holding the called routine's name in a variable may be  
10 analyzed by data flow analysis to determine the potential values of the target called routine.  
11 Dynamic information obtained during the execution of the COBOL dynamic CALL may  
12 augment the static information obtained through the static analysis and data flow analysis.  
13

14 One aspect of a preferred embodiment of the present invention identifies computer  
15 program code which is a candidate meeting a query constraint.  
16

17 Another aspect of a preferred embodiment of the present invention identifies computer  
18 program code which is a candidate for a stored procedure.  
19

20 Another aspect of a preferred embodiment of the present invention identifies computer  
21 program code which is a candidate for Web enablement.  
22

23 Another aspect of a preferred embodiment of the present invention scans and parses  
24 source code corresponding to computer program code to determine static information  
25 concerning the computer program code, and stores the static information in a database.  
26

27 Another aspect of a preferred embodiment of the present invention collects dynamic  
28 information concerning the computer program code during an execution of the computer  
29 program code, and stores the dynamic information in the database.

1 Another aspect of a preferred embodiment of the present invention discovers or  
2 develops relationships and dependencies responsive to the static information and dynamic  
3 information stored in the database, and stores the relationships and dependencies in the  
4 database.

5  
6 Another aspect of a preferred embodiment of the present invention queries the database  
7 to produce a set of potential candidates of computer program code meeting a constraint of the  
8 query. If insufficient candidates are returned by the query, then the query constraint may be  
9 relaxed, and the query repeated.

10  
11 Another aspect of a preferred embodiment of the present invention identifies a  
12 candidate for re-implementation as a database stored procedure call by a query searching the  
13 database for a portion of the computer program code having static information indicating that  
14 the portion of the computer program code contains a number, above a specified first threshold,  
15 of calls to a database management system, and having dynamic information indicating that the  
16 portion of the computer program code is subject to a number of calls, above a specified second  
17 threshold, by another portion of computer program code.

18  
19 Another aspect of a preferred embodiment of the present invention identifies a  
20 candidate for re-implementation as a Web-enabling interface call by a query searching the  
21 database for a portion of the computer program code having static information indicating that  
22 the portion of the computer program contains a transaction and does not contain screen output  
23 related program code.

1 A preferred embodiment of the present invention has the advantage of promoting reuse  
2 of an existing application program's business logic, resources, or other assets.

3  
4 A preferred embodiment of the present invention has the advantage of promoting  
5 impact analysis and application understanding of an existing application program.

6  
7 A preferred embodiment of the present invention has the further advantage of  
8 promoting reuse of an existing application program's business logic to link the existing  
9 business logic with the Web via a connector to form a new Web application program.

10  
11 A preferred embodiment of the present invention has the further advantage of providing  
12 improved analysis, identification, and isolation of the business logic of an application program.

13  
14 A preferred embodiment of the present invention has the further advantage of reducing  
15 or eliminating labor intensive efforts in the creation of a Web connector through the analysis,  
16 identification, and isolation of the business logic.

17  
18 A preferred embodiment of the present invention has the further advantage of  
19 promoting the use of an enterprise's existing programming skills.

20  
21 A preferred embodiment of the present invention has the further advantage of providing  
22 an application knowledge base, a database containing static information, dynamic information,  
23 and relationships and dependencies of computer program code, which may be queried to  
24 identify candidate computer program code.

25  
26 A preferred embodiment of the present invention has the further advantage of providing  
27 identification of computer program code which is a candidate for stored procedures.

28  
29 A preferred embodiment of the present invention has the further advantage of providing



1 identification of computer program code which is a candidate for Web enablement.

2

3 A preferred embodiment of the present invention has the further advantage of providing  
4 an improved user interface for identification of computer program code which is a candidate  
5 for Web enablement or stored procedures.

6

7 A preferred embodiment of the present invention has the further advantage of providing  
8 improved usability in a tool for identification of computer program code which is a candidate  
9 for Web enablement or stored procedures.

10

11 A preferred embodiment of the present invention has the further advantage of providing  
12 improved functionality in a tool for identification of computer program code which is a  
13 candidate for Web enablement or stored procedures.

14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
21

## BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the Description of the Preferred Embodiment in conjunction with the attached Drawings, in which:

**Figure 1** is a block diagram of a distributed computer system used in performing the method of the present invention, forming part of the apparatus and computer program product of the present invention, and which may use the article of manufacture and computer program product comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to practice the present invention;

**Figure 2** is a block diagram of a preferred embodiment of the present invention;

**Figure 3** illustrates a preferred embodiment of the static scanner or parser portion of the preferred embodiment of the present invention;

**Figure 4** illustrates a preferred embodiment of the dynamic scanner or execution monitor parser portion of the preferred embodiment of the present invention;

**Figure 5** illustrates a preferred embodiment of the relationship analyzer portion of the preferred embodiment of the present invention; and

**Figure 6** is a flowchart illustrating the operations preferred in carrying out the preferred embodiment of the present invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. A person skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will be apparent to a person skilled in the relevant art that this invention can also be employed in a variety of other devices and applications.

Referring first to **Figure 1**, there is depicted a graphical representation of a data processing system **8**, which may be utilized to implement the present invention. As may be seen, data processing system **8** may include a plurality of networks, such as Local Area Networks (LAN) **10** and **32**, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Alternatively, networks **10** and **32** may be intranets or portions of the internet. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network. Each said network may also consist of a plurality of processors coupled via a communications medium, such as shared memory, shared storage, or an interconnection network. As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16** and may be provided with a pointing device such as a mouse **17**.

The data processing system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be preferably coupled to LAN **10** by means of communications link **22**. The mainframe computer **18** may also be coupled to a storage device **20** which may serve as remote storage for LAN **10**. Similarly, LAN **10** may be coupled via communications link **24** through a sub-system control unit/communications controller **26** and

1 communications link **34** to a gateway server **28**. The gateway server **28** may be an IWS which  
2 serves to link LAN **32** to LAN **10**. Preferably, server **28** is a web application server which  
3 passes transactions from a requester **30** on the internet **32** to the mainframe **18** upon which a  
4 back-end application serving the transaction is executing.

5  
6 With respect to LAN **32** and LAN **10**, a plurality of documents or resource objects may  
7 be stored within storage device **20** and controlled by mainframe computer **18**, as resource  
8 manager or library service for the resource objects thus stored. Of course, those skilled in the  
9 art will appreciate that mainframe computer **18** may be located a great geographic distance  
10 from LAN **10** and similarly, LAN **10** may be located a substantial distance from LAN **32**. For  
11 example, LAN **32** may be located in California while LAN **10** may be located within North  
12 Carolina and mainframe computer **18** may be located in New York.

13  
14 Software program code which employs the present invention is typically stored in the  
15 memory of a storage device **14** of a stand alone workstation or LAN server from which a  
16 developer may access the code for distribution purposes, the software program code may be  
17 embodied on any of a variety of known media for use with a data processing system such as a  
18 diskette or CD-ROM or may be distributed to users from a memory of one computer system  
19 over a network of some type to other computer systems for use by users of such other systems.  
20 Such techniques and methods for embodying software code on media and/or distributing  
21 software code are well-known and will not be further discussed herein.

22  
23 As will be appreciated upon reference to the foregoing, it is often desirable for a user  
24 to link an application program on the mainframe **18** to the internet **32** and/or World Wide Web  
25 (Web), where the application program was not originally designed for Web or internet based  
26 transactions. For such Web-enablement, the user needs to identify call interfaces of the  
27 application program which conventionally is a manual process. If this application program has  
28 a database dependency, then it may also be desirable for this user to identify call interfaces or  
29 computer program code which may be re-implemented as a database stored procedure call. A

1 preferred embodiment of the present invention assists a user in performing such an  
2 identification of computer program code which is a candidate for a particular purpose, such as  
3 re-implementation. Alternative embodiments identify candidates for re-implementation such as  
4 Web enablement or a stored procedure.  
5

6 Referring now to **Figure 2** illustrating a block diagram of a preferred embodiment of  
7 the present invention, the preferred embodiment comprises a database **215** (Application  
8 Knowledge Base) for storing static information **220**, dynamic information **225**, and  
9 relationships and dependencies **250** corresponding to computer code in application inventory  
10 **205**. The database in the preferred embodiment is an IBM® DB2® UDB (Universal  
11 Database); however, other relational or non-relational (object, network, hierarchal, flat, . . . )  
12 databases may be used in addition to either persistent or in-memory databases without  
13 departing from the spirit and scope of the present invention. (IBM® and DB2® are registered  
14 trademarks of International Business Machines Corporation in the United States, other  
15 countries, or both.) The static information **220** concerning the computer program code **205** is  
16 determined by a scanner, parser, or other software analysis tool **210**. An execution monitor or  
17 scanner **230** collects dynamic information **225** concerning the computer program code **205**  
18 during an execution **235** of the computer program code. The static information **220** and the  
19 dynamic information **225** are analyzed by an relationship analyzer **245** to discover and/or  
20 develop relationships and dependencies **250** responsive to the static information **220** and  
21 dynamic information **225**. A query **255** of the Application Knowledge Base **215** may then  
22 produce a set of potential candidates **260** of computer program code meeting a constraint of the  
23 query. The query facility of the preferred embodiment is IBM net.Data, a dynamic HTML  
24 (HyperText Markup Language) page generator; however, other query facilities may be used  
25 such as a Java servlet or a Java data bean. (Java and all Java-based trademarks are trademarks  
26 of Sun Microsystems, Inc. in the United States, other countries, or both.)  
27

28 **Figure 3** illustrates a preferred embodiment **300** of the static scanner, parser, or other  
29 analysis tool **210** which collects the static information **220** concerning the computer program

code **205**. The static scanner **300** may comprise multiple static analysis tools. In the preferred embodiment, the static scanner **300** comprises a control/environment parsing tool **305**, a transaction parsing tool **310**, a program source parsing tool **315**, a database parsing tool **320**, and an executable/ load module parsing tool **325**. The preferred control/environment parsing tool **305** is a JCL (Job Control Language) parsing tool which analyzes JCL **330** to provide program execution, dataset creation and use, and database use **335**. An IBM Customer Information Control System (CICS®) parsing tool and an IBM Information Management System (IMS) parsing tool are preferred transaction parsing tools **310** providing program execution, dataset and database use **340** from analyzing transaction definitions and file definitions **345**. (CICS® is a registered trademark of International Business Machines Corporation in the United States, other countries, or both.) Application program source **350** may be analyzed by the program source parsing tool **315** to provide data record definitions, data store I/O, call relationships, and data element use **355**. A database parsing tool for IMS DBD (Data Base Definition) and IMS PSB (Program Specification Block) and a database parsing tool to analyze DB2 table information are preferred for the database parsing tool **320** which analyzes database definitions **360** to provide linkages **365** between the application program source and databases such as IMS, PSB, and DB2 among others. The executable/load module parsing tool **325** analyzes load modules **370** to provide module, CSECT (control section), translator, and translation date **375**.

**Figure 4** illustrates a preferred embodiment **400** of the execution monitor or dynamic scanner **230** which collects dynamic information **225** concerning the computer program code **205** during an execution **235** of the computer program code. The dynamic scanner **400** may also comprise multiple dynamic analysis tools, comprising in the preferred embodiment a coverage and performance measurement tool **410** and an inter-dependency tool **420**. The coverage and performance measurement tool **410** analyzes the execution **430** of an application program run unit and compilation unit to determine the coverage for single or multiple test cases **440** and to measure performance **450**. A preferred inter-dependency tool **420** is a CICS Inter-dependencies Utility which dynamically captures resource usage, transaction flow **460**

1 from a CICS region **470**. The execution monitor in the preferred embodiment is an IBM  
2 CICS Affinities Utility and the IBM CICS Interdependency Utility; however, those skilled in  
3 the art will recognize other execution monitors may be used without departing from the spirit  
4 and scope of the invention.

5  
6 **Figure 5** illustrates a preferred embodiment **500** of the relationship analyzer **245** which  
7 discovers and/or develops relationships and dependencies **250** responsive to the static  
8 information **220** and dynamic information **225**. In the preferred embodiment, the relationship  
9 analyzer **500** provides automatic, generalized, inter-relationship analysis by performing  
10 IMS DL/I (Information Management System Data Language 1) call parameter analysis **510**,  
11 call hierarchy expansion **520**, load module to source relationship mapping **530**, CICS  
12 pseudo-conversation flow analysis **540**, missing and duplicate component analysis **550**, and un-  
13 referenced component analysis **560**.

14  
15 Referring now to **Figure 6**, the flowchart illustrates the operations preferred in carrying  
16 out the preferred embodiment of the present invention. In the flowcharts, the graphical  
17 conventions of a diamond for a test or decision and a rectangle for a process or function are  
18 used. These conventions are well understood by those skilled in the art, and the flowcharts are  
19 sufficient to enable one of ordinary skill to write code in any suitable computer programming  
20 language. The process **600** of identifying computer program code which is a candidate for a  
21 particular purpose, such as re-implementation for Web enablement or stored procedures begins  
22 at process block **605**. Process block **610** then uses static scanner **210** to scan or parse the  
23 application inventory **205** retaining static information **220** concerning the artifacts of the  
24 application program. Thereafter, process block **615** stores the static information **220** in the  
25 Application Knowledge Base **215**. Process block **620** augments the Application Knowledge  
26 Base **215** with dynamic information **225** obtained by an execution scanner or monitor **230**  
27 which collects the dynamic information **225** during the execution of an application program  
28 executable **235** produced by an assembler, compiler, or interpreter **240** from the computer  
29 program source code. After collection of the static information **220** and dynamic information

1     **225**, process block **625** uses relationship analyzer **245** to discover and develop relationships  
2     and dependencies **250** within the Application Knowledge Base **215**, and then stores these  
3     relationships and dependencies **250** in the Application Knowledge Base **215**. Thereafter,  
4     process block **630** may query **255** the application knowledge base **215** to produce a set of  
5     potential candidates **260** of computer program code meeting a constraint of the query.  
6

7             For example, to determine which applications may benefit by being re-implemented as  
8     a database stored procedure, the query may search for routines containing SQL (Structured  
9     Query Language) statements that are the targets of a number of CALLs where the number is  
10    above a specified threshold. Thus the present invention provides a capability of identifying  
11    more likely candidates as opposed to the conventional manual process of looking for routines  
12    that contain a large amount of SQL. To determine which application programs are candidates  
13    for a Web interface under CICS, the query may search for CICS transactions that contain a  
14    COMMAREA (communication area) linkage and that are void of BMS (Basic Mapping  
15    Support) MAP send and/or receive statements (i.e., that lack screen I/O (input output)).  
16

17            After the results of the query are returned, decision block **635** determines if sufficient  
18    candidates are identified and returned. If a sufficient number of candidates are not discovered,  
19    then process block **645** allows relaxing the query constraints, after which control returns to  
20    process block **630** to query the application knowledge base **215** with the relaxed constraints.  
21

22            Returning now to decision block **635**, if sufficient candidates are identified, the process  
23    may end at process block **640**.  
24

25            Using the foregoing specification, the invention may be implemented using standard  
26    programming and/or engineering techniques using computer programming software, firmware,  
27    hardware or any combination or sub-combination thereof. Any such resulting program(s),  
28    having computer readable program code means, may be embodied within one or more  
29    computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,



1 semiconductor memories such as Read-Only Memory (ROM), Programmable Read-Only  
2 Memory (PROM), etc., or any memory or transmitting device, thereby making a computer  
3 program product, i.e., an article of manufacture, according to the invention. The article of  
4 manufacture containing the computer programming code may be made and/or used by  
5 executing the code directly or indirectly from one medium, by copying the code from one  
6 medium to another medium, or by transmitting the code over a network. An apparatus for  
7 making, using, or selling the invention may be one or more processing systems including, but  
8 not limited to, central processing unit (CPU), memory, storage devices, communication links,  
9 communication devices, servers, input/output (I/O) devices, or any sub-components or  
10 individual parts of one or more processing systems, including software, firmware, hardware or  
11 any combination or sub-combination thereof, which embody the invention as set forth in the  
12 claims.

13  
14 User input may be received from the keyboard, mouse, pen, voice, touch screen, or any  
15 other means by which a human can input data to a computer, including through other programs  
16 such as application programs.

17  
18 One skilled in the art of computer science will easily be able to combine the software  
19 created as described with appropriate general purpose or special purpose computer hardware to  
20 create a computer system and/or computer sub-components embodying the invention and to  
21 create a computer system and/or computer sub-components for carrying out the method of the  
22 invention. Although the present invention has been particularly shown and described with  
23 reference to a preferred embodiment, it should be apparent that modifications and adaptations  
24 to that embodiment may occur to one skilled in the art without departing from the spirit or  
25 scope of the present invention as set forth in the following claims.

## CLAIMS

We claim:

1. A method of identifying a portion of computer program code as a candidate for a modification, said method comprising the steps of :
  - scanning and parsing source code corresponding to the computer program code to determine static information concerning the computer program code;
  - storing the static information in a database;
  - collecting dynamic information concerning the computer program code during an execution of the computer program code;
  - storing the dynamic information in the database;
  - developing relationships and dependencies responsive to the static information and dynamic information stored in database;
  - storing the relationships and dependencies in the database; and
  - providing for a query of the database to produce a set of potential candidates of computer program code meeting a constraint of the query.
2. The method of claim 1 wherein if insufficient candidates are returned by the query, then providing for a relaxing of the constraint, and a repetition of the query.
3. The method of claim 2 further comprising the step of identifying a portion of the computer program code as a candidate for re-implementation as a database stored procedure call.

1 4. The method of claim 3 wherein the query searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program code contains a call to a database management system, and having dynamic  
4 information indicating that the portion of the computer program code is subject to a call by  
5 another portion of computer program code.

1 5. The method of claim 4 wherein the query searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program code contains a number, above a specified first threshold, of calls to a database  
4 management system, and having dynamic information indicating that the portion of the  
5 computer program code is subject to a number of calls, above a specified second threshold, by  
another portion of computer program code.

6. The method of claim 2 further comprising the step of identifying a portion of the  
computer program code as a candidate for re-implementation as a Web-enabling interface call.

7. The method of claim 6 wherein the query searches the database for a portion of the  
computer program code having static information indicating that the portion of the computer  
program code contains a transaction call.

1 8. The method of claim 7 wherein the query also searches the database for a portion of the  
2 computer program code having static information indicating that the portion of the computer  
3 program does not contain screen output related program code.

1 9. An article of manufacture for use in a computer system for identifying a portion of  
2 computer program code as a candidate for a modification, said article of manufacture  
3 comprising a computer-readable storage medium having a computer program embodied in said  
4 medium which causes the computer system to execute a method comprising the steps of:

5 scanning and parsing source code corresponding to the computer program code  
6 to determine static information concerning the computer program code;

7 storing the static information in a database;

8 collecting dynamic information concerning the computer program code during  
9 an execution of the computer program code;

10 storing the dynamic information in the database;

11 developing relationships and dependencies responsive to the static information  
12 and dynamic information stored in database;

13 storing the relationships and dependencies in the database; and

14 providing for a query of the database to produce a set of potential candidates of  
15 computer program code meeting a constraint of the query.

16 10. The article of manufacture of claim 9 wherein if insufficient candidates are returned by  
17 the query, then providing for a relaxing of the constraint, and a repetition of the query.

1 11. The article of manufacture of claim 10 wherein the embodied computer program  
2 causes the computer system to execute the method step of identifying a portion of the  
3 computer program code as a candidate for re-implementation as a database stored procedure  
4 call.

1 12. The article of manufacture of claim 11 wherein the query searches the database for a  
2 portion of the computer program code having static information indicating that the portion of  
3 the computer program code contains a call to a database management system, and having  
4 dynamic information indicating that the portion of the computer program code is subject to a  
5 call by another portion of computer program code.

1 13. The article of manufacture of claim 12 wherein the query searches the database for a  
2 portion of the computer program code having static information indicating that the portion of  
3 the computer program code contains a number, above a specified first threshold, of calls to a  
4 database management system, and having dynamic information indicating that the portion of  
5 the computer program code is subject to a number of calls, above a specified second threshold,  
6 by another portion of computer program code.

1 14. The article of manufacture of claim 10 wherein the embodied computer program  
2 causes the computer system to execute the method step of identifying a portion of the  
3 computer program code as a candidate for re-implementation as a Web-enabling interface call.

1 15. The article of manufacture of claim 14 wherein the query searches the database for a  
2 portion of the computer program code having static information indicating that the portion of  
3 the computer program code contains a transaction call.

1 16. The article of manufacture of claim 15 wherein the query also searches the database for  
2 a portion of the computer program code having static information indicating that the portion of  
3 the computer program does not contain screen output related program code.

1 17. A computer system for identifying a portion of computer program code as a candidate  
2 for a modification, said computer system comprising :

3 a parser for scanning and parsing source code corresponding to the computer  
4 program code to determine static information concerning the computer program code;

5 an execution monitor for collecting dynamic information concerning the  
6 computer program code during an execution of the computer program code;

7 relationships and dependencies developed responsive to the static information  
8 and dynamic information;

9 a database for storing the static information, the dynamic information, and the  
10 relationships and dependencies; and

11 a query of the database to produce a set of potential candidates of computer  
12 program code meeting a constraint of the query.

1 18. The computer system of claim 17 wherein if insufficient candidates are returned by the  
2 query, then providing for a relaxing of the constraint, and a repetition of the query.

1 19. The computer system of claim 18 further comprising an identification of a portion of  
2 the computer program code as a candidate for re-implementation as a database stored procedure  
3 call.

1 20. The computer system of claim 19 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a call to a database management system, and having dynamic  
4 information indicating that the portion of the computer program code is subject to a call by  
5 another portion of computer program code.

1 21. The computer system of claim 20 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a number, above a specified first threshold, of calls to a  
4 database management system, and having dynamic information indicating that the portion of  
5 the computer program code is subject to a number of calls, above a specified second threshold,  
6 by another portion of computer program code.

1 22. The computer system of claim 18 further comprising an identification of a portion of  
2 the computer program code as a candidate for re-implementation as a Web-enabling interface  
3 call.

1 23. The computer system of claim 22 wherein the query searches the database for a portion  
2 of the computer program code having static information indicating that the portion of the  
3 computer program code contains a transaction call.

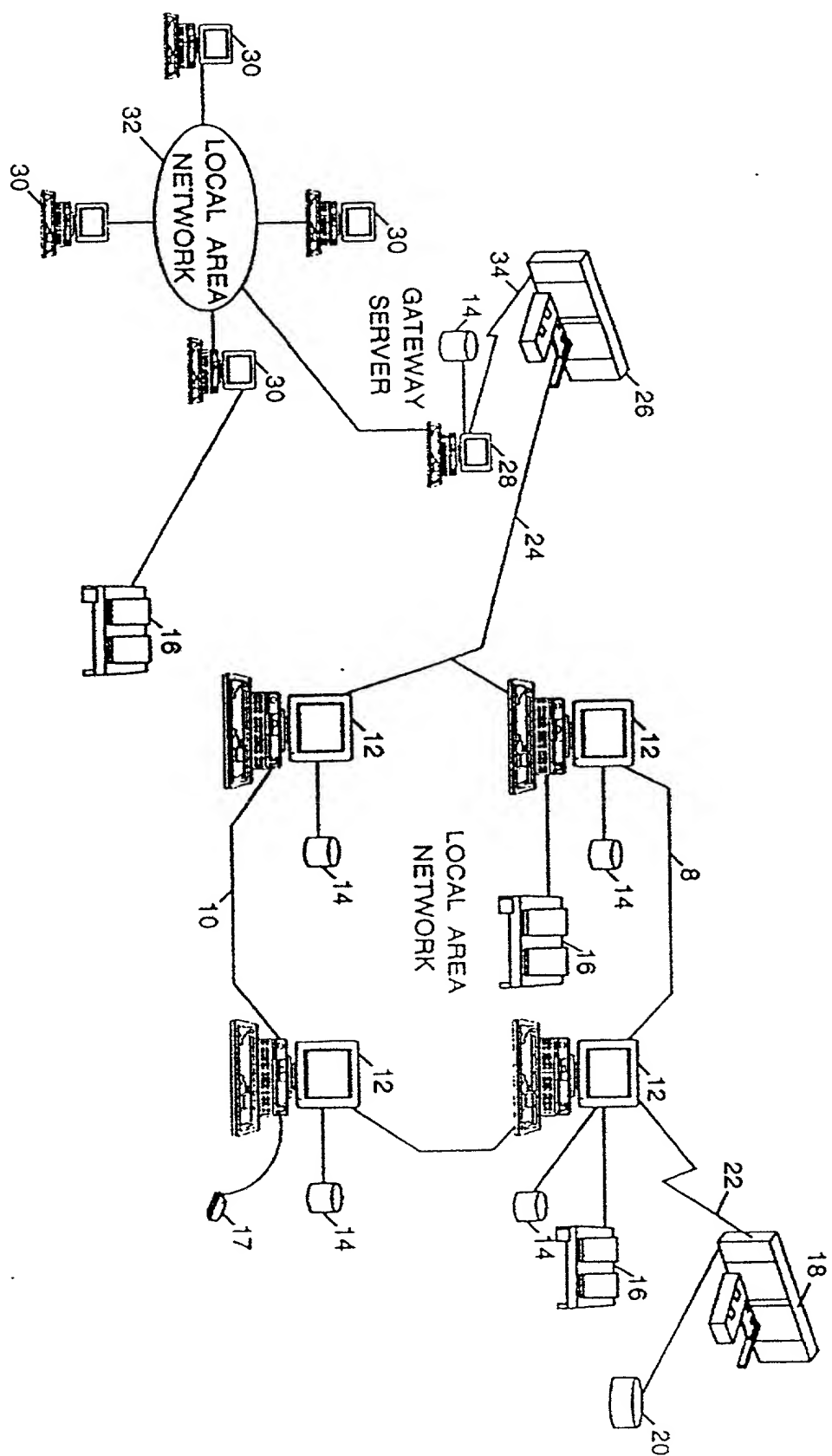
1 24. The computer system of claim 23 wherein the query also searches the database for a  
2 portion of the computer program code having static information indicating that the portion of  
3 the computer program does not contain screen output related program code.

## ABSTRACT

### **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING AUTOMATIC IDENTIFICATION OF A COMPUTER PROGRAM CODE CANDIDATE FOR WEB DEPLOYMENT OR A STORED PROCEDURE**

Computer program code which is a candidate for Web enablement or stored procedures is identified. Source code corresponding to computer program code is scanned and parsed to determine static information concerning the computer program code. The static information is stored in a database. Dynamic information concerning the computer program code during an execution of the computer program code is also collected and stored in the database. Responsive to the static information and dynamic information stored in the database, relationships and dependencies are then developed and stored in the database. The database may then be queried to produce a set of potential candidates of computer program code meeting a constraint of the query. If insufficient candidates are returned by the query, then the query constraint may be relaxed, and the query repeated. A candidate for re-implementation as a database stored procedure call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program code contains a number, above a specified first threshold, of calls to a database management system, and having dynamic information indicating that the portion of the computer program code is subject to a number of calls, above a specified second threshold, by another portion of computer program code. A candidate for re-implementation as a Web-enabling interface call may be identified by a query searching the database for a portion of the computer program code having static information indicating that the portion of the computer program contains a transaction and does not contain screen output related program code.





**FIG. 1**

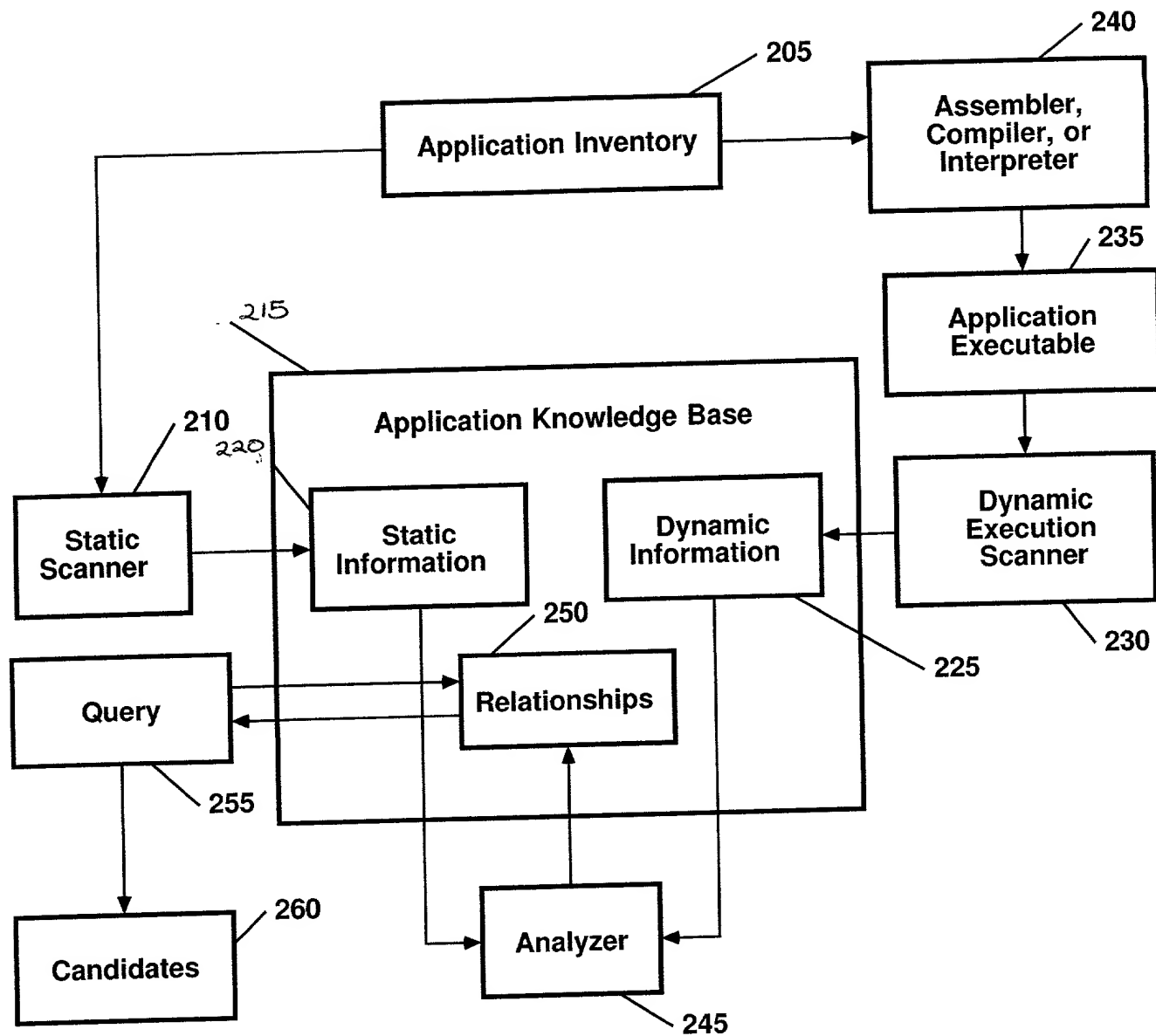
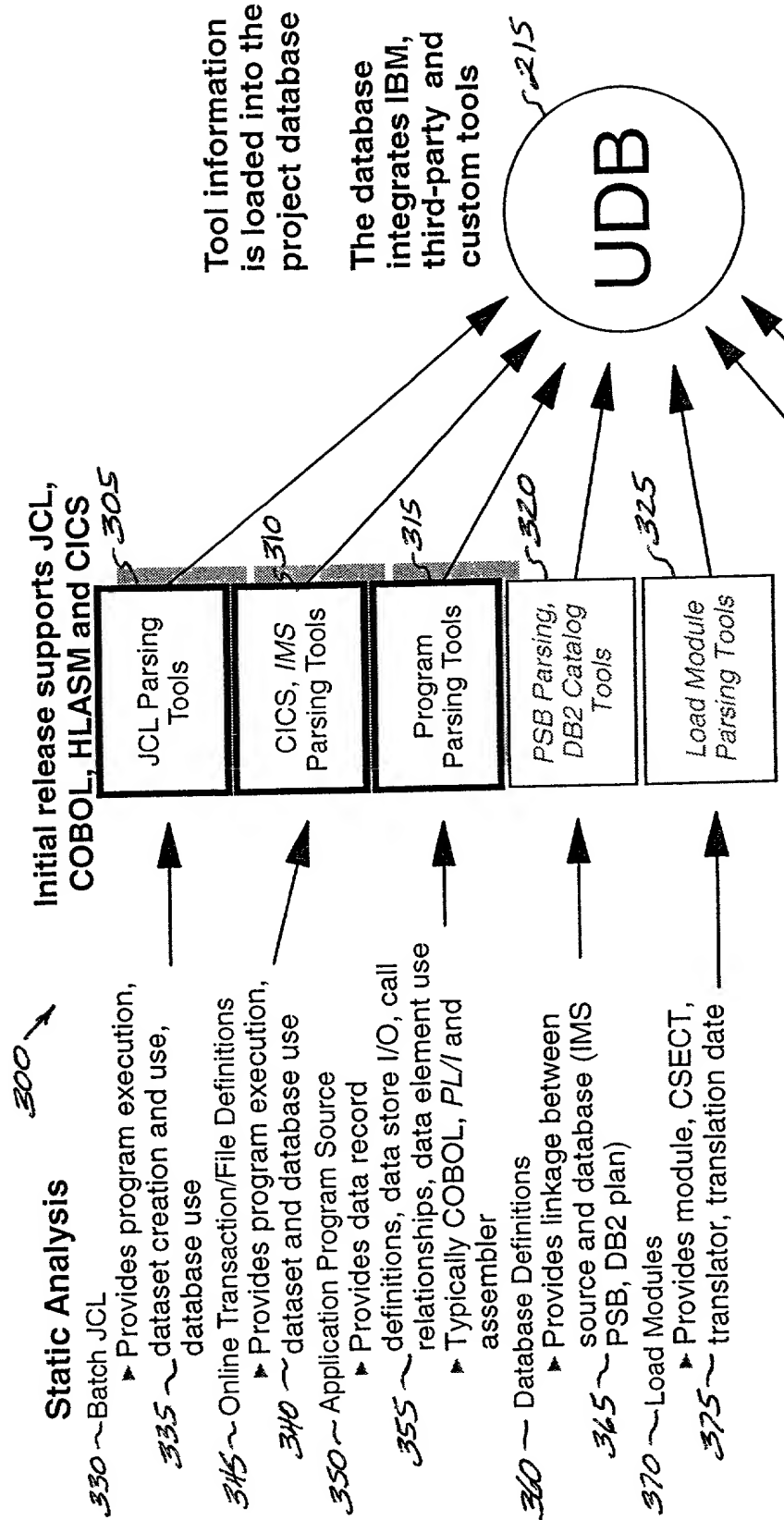


Fig. 2



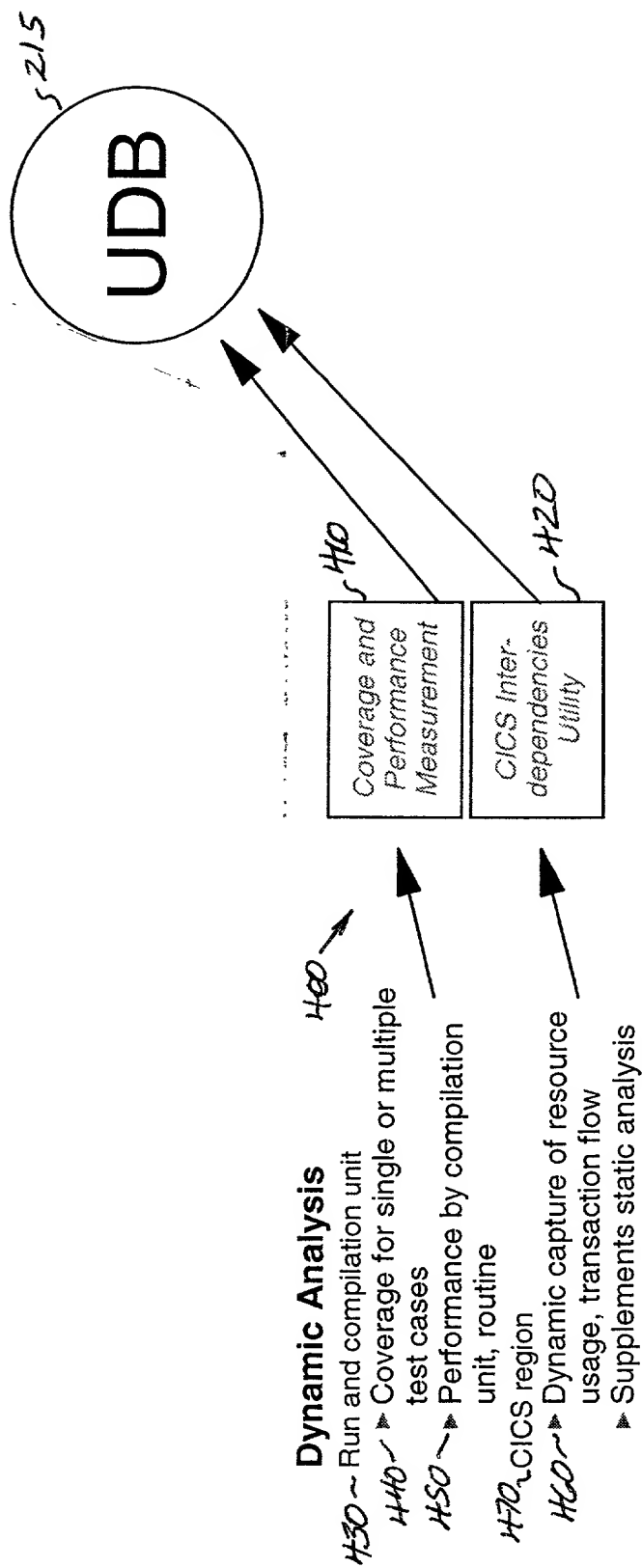
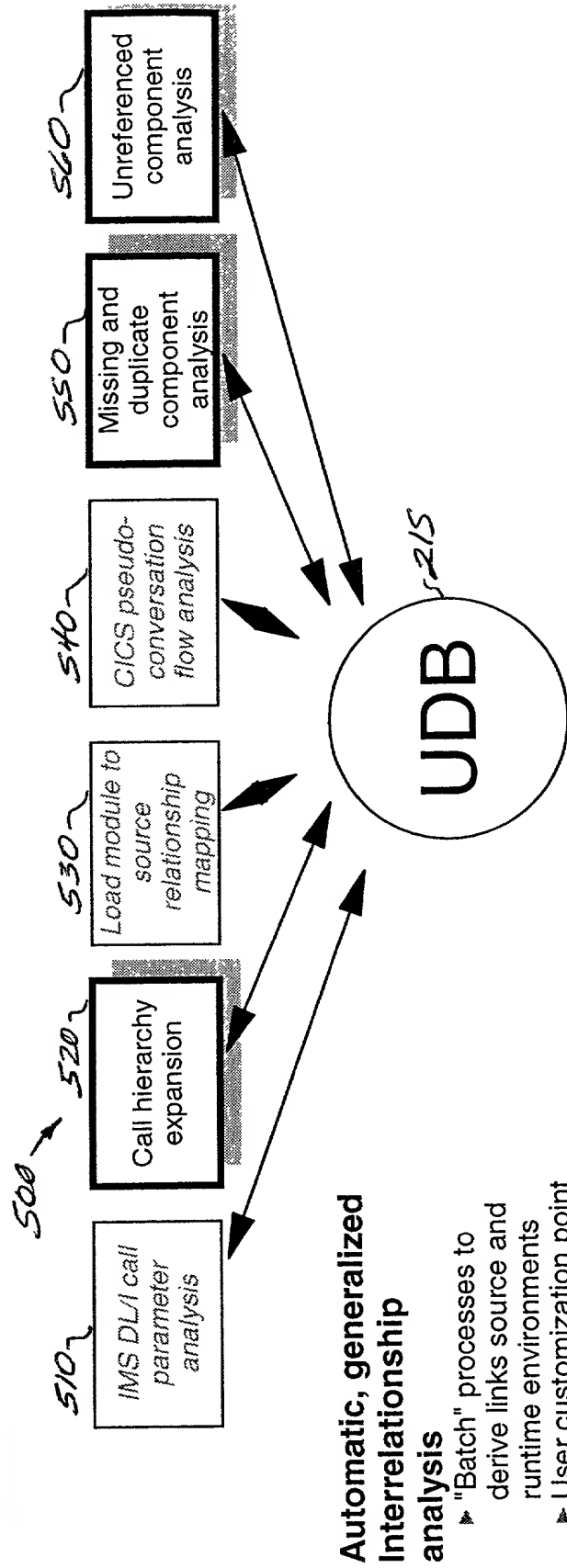


Fig. 4



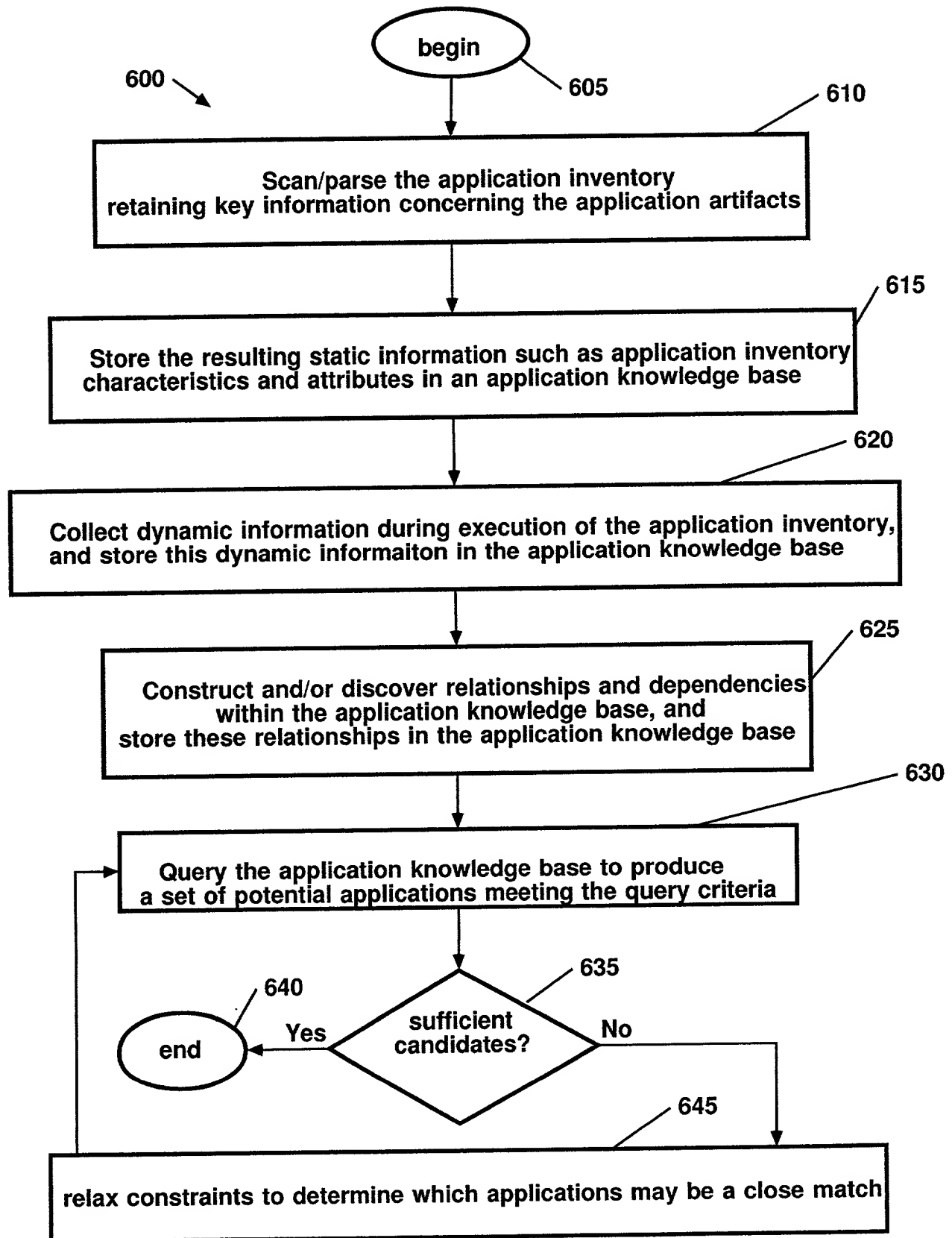


Fig. 6

## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0080US1

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method of, System for, and Computer Program Product for Providing Automatic Identification of a Computer Program Code Candidate for Web Deployment or a Stored Procedure

the specification of which (check one)

☒ is attached hereto.  
☐ was filed on \_\_\_\_\_  
as Application Serial No. \_\_\_\_\_  
and was amended on \_\_\_\_\_ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)	Priority Claimed
<input checked="" type="checkbox"/> None	<input type="checkbox"/> Yes <input type="checkbox"/> No
(Number)	(Country) (Day/Month/Year Filed)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

None		
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0080US1

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Romualdas Strimaitis (#35,697)	Christopher A. Hughes (#26,914)
Timothy M. Farrell (#37,321)	Edward A. Pennington (#32,588)
Ingrid M. Foerster (#36,511)	John E. Hoel (#26,279)
Prentiss W. Johnson (# 33,123)	Joseph C. Redmond, Jr. (#18,753)

**Send correspondence to:**

Prentiss Wayne Johnson  
Attorney  
IBM Corporation  
Intellectual Property Law  
555 Bailey Avenue (J46/G467)  
San Jose CA 95141-1003

**Direct Telephone Calls to:** (name and telephone number) Prentiss W. Johnson, 408-463-5673

**Full name of sole or first joint-inventor:** Curt L. Cotner

Inventor's signature:

Date:

Residence: 1975 Day Road, Gilroy, California 95020

Citizenship: United States of America

Post Office Address: Same

**Full name of second joint-inventor:** Laurence E. England

Inventor's signature:

Date:

Residence: 520 La Canada Court, Morgan Hill, California 95037

Citizenship: United States of America

Post Office Address: Same

**Full name of third joint-inventor:** Howard J. Glaser

Inventor's signature:

Date:

Residence: 5808 Vargas Court, San Jose, California 95120

Citizenship: United States

Post Office Address: Same



DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0080US1

Full name of fourth joint-inventor: Howard M. Hess

Inventor's signature:

Date:

Residence: 3033 Thayer Street, Evanston, IL 60201-1146

Citizenship: United States

Post Office Address: Same

11/11/2000 11:11 AM  
STL9-2000-0080US1  
Howard M. Hess  
3033 Thayer Street  
Evanston, IL 60201-1146  
United States  
Same